Consultative Committee for Space Data Systems

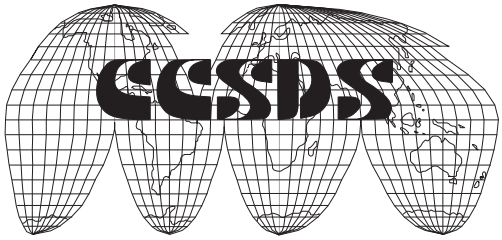
EXPERIMENTAL SPECIFICATION FOR SPACE DATA SYSTEM STANDARDS

Low Density Parity Check Code Family

CCSDS 0.0-1-0

ORANGE BOOK

April 2006 [AR4JA Codes]



AUTHORITY

Issue:	Orange Book, Issue 0.1
Date:	April 20, 2006
Location:	Pasadena, California

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and reflects the consensus of technical panel experts from CCSDS Member Agencies. The procedure for review and authorization of CCSDS Reports is detailed in the *Procedures Manual for the Consultative Committee for Space Data Systems.*

This document is published and maintained by:

CCSDS Secretariat
Office of Space Communication (Code M-3)
National Aeronautics and Space Administration
Washington, DC 20546, USA

PREFACE

This document is a CCSDS Experimental Specification. Its Experimental status indicates that it is part of a research or development effort based on prospective requirements, and as such it is not considered a Standards Track document. Experimental Specifications are intended to demonstrate technical feasibility in anticipation of a 'hard' requirement that has not yet emerged. Experimental work may be rapidly transferred onto the Standards Track should a hard requirement emerge in the future.

FOREWORD

This Experimental Specification describes end-to-end resource provisioning for orbiting missions within the proposed Next Generation Space Internet (NGSI) architecture.

Through the process of normal evolution, it is expected that expansion, deletion, or modification to this document may occur. This Experimental Specification is therefore subject to CCSDS document management and change control procedures which are defined in the *Procedures Manual for the Consultative Committee for Space Data Systems*. Current versions of CCSDS documents are maintained at the CCSDS Web site:

http://www.ccsds.org/

Questions relating to the contents or status of this report should be addressed to the CCSDS Secretariat at the address on page i.

At time of publication, the active Member and Observer Agencies of the CCSDS were:

Member Agencies

- Agenzia Spaziale Italiana (ASI)/Italy.
- British National Space Centre (BNSC)/United Kingdom.
- Canadian Space Agency (CSA)/Canada.
- Centre National d'Etudes Spatiales (CNES)/France.
- Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- National Aeronautics and Space Administration (NASA)/USA.
- National Space Development Agency of Japan (NASDA)/Japan.
- Russian Space Agency (RSA)/Russian Federation.

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- Centro Tecnico Aeroespacial (CTA)/Brazil.
- Chinese Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- Communications Research Centre (CRC)/Canada.
- Communications Research Laboratory (CRL)/Japan.
- Danish Space Research Institute (DSRI)/Denmark.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Federal Service of Scientific, Technical & Cultural Affairs (FSST&CA)/Belgium.
- Hellenic National Space Committee (HNSC)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space and Astronautical Science (ISAS)/Japan.
- Institute of Space Research (IKI)/Russian Federation.
- KFKI Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- MIKOMTEK: CSIR (CSIR)/Republic of South Africa.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- National Oceanic & Atmospheric Administration (NOAA)/USA.
- National Space Program Office (NSPO)/Taipei.
- Space & Upper Atmosphere Research Commission/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.
- United States Geological Survey (USGS)/USA.

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS-01	Low Density Parity Check Code Family	April, 2006	Revision 01
CCSDS-00	Low Density Parity Check Code Family	January, 2006	First release

Contents

1 Background **1**

2 Introduction **2**

3 Specification **2**

4 Parity check matrices **3**
4.1 Rate 1/2, 2/3, 3/4, and 4/5 codes 3

5 Randomization and Synchronization **4**

6 Encoders **8**
6.1 Encoding Example : Rate-1/2 $k = 32$ 8
6.2 Encoding Example : Rate-4/5 $k = 1024$ 10

A Performance **12**

List of Figures

1 An H Matrix for the $(n = 1280, k = 1024)$ rate-4/5 code. For this code M=128 and columns 1281 through 1408 are punctured. 4
2 A Quasicyclic Encoder Using Feedback Shift Registers 8
3 Protograph of rate 1/2 AR4JA LDPC code. 9
4 Performance of LDPC Codes: red=rate-1/2, green=rate-2/3, blue=rate-4/5, for $k = 16384, 4096, 1024$. 13

1 Background

The family of LDPC codes presented in this document are intended to complement the current codes in the CCSDS Telemetry Channel Coding Blue Book [1], and were designed according to a list of requirements and evaluation criteria that reflect the needs of spacecraft applications [5].

Requirements

1. *Code rates:* The family shall include codes of rate $\approx 1/2$ and $\approx 7/8$
2. *Block lengths:* The family shall cover k ≈ 1000 to k ≈ 16000 information bits spaced by multiples of ≈ 4
3. *Family:* A single hardware decoder shall be appropriate for all codes
4. *Intellectual property:* There must be no restrictions for CCSDS members

Desired Properties

1. *Systematic encoders:* Systematic encoders are perferred
2. *Code rates:* One or two intermediate rates from 1/2, 2/3, 3/4, 5/6, 6/7 are desired

Evaluation Criteria

1. *Decoder computation:* Codes requiring fewer decoder message computations are preferred
2. *Encoder computation:* Preferred encoders require fewer logic gates for a given speed
3. *Descriptional complexity:* The code description in a standards document should be short
4. *Code performance:* Codes requiring less E_b/N_0 at WER=10^{-4} and 10^{-6} are preferred

The selected code rates are 1/2, 2/3, and 4/5, three values which are about uniformly spaced by 1 dB on the rate-dependent capacity curve for the binary-input AWGN channel [7]. Near rate 1/2, a 1% improvement in bandwidth efficiency costs about 0.02 dB in power efficiency; near rate 7/8, a 1% improvement in bandwidth efficiency costs 0.1 dB in power efficiency. Hence, the use of a higher order modulation may be a more practical means for saving bandwidth than the use of a code with rate much above 0.8. The code rates are exact ratios of small integers to simplify implementation.

The selected block lengths are $k = 1024$, $k = 4096$, and $k = 16384$. The three values $k = \{1024, 4096, \infty\}$ are about uniformly spaced by 0.6 dB on the sphere-packing bound at word error rate of 10^{-8}, and reducing the last value from ∞ to 16384 makes the largest block size practical at a cost of about 0.3 dB. By choosing to keep k constant among family members, rather than n, the spacecraft's command and data handling system can generate data frames without knowledge of the code rate. Choosing powers of 2 may simplify implementation.

Implementers should be aware that many patents have been filed on LDPC codes; in particular, a procedure for parallelized decoding of LDPC codes is covered by a US patent granted to T. Richardson and V. Novichkov, "Methods and Apparatus for Decoding LDPC Codes," United States Patent No. US 6,633,856 B2, Oct. 14, 2003.

The selected codes are systematic. Two low-complexity encoding methods are described [6], and either can be used, depending on constraints of the chosen technology (software,

FPGA, ASIC). The parity check matrices have plenty of structure to facilitate decoder implementation [9]. The codes have irregular degree distributions, because this improves performance by about 0.5 dB at rate 1/2, compared to a regular $(3, 6)$ code [2][3][4].

2 Introduction

Like turbo codes, low-density parity-check (LDPC) codes are binary block codes with large code blocks (hundreds or thousands of bits). They may be systematic or non-systematic, and they may be transparent or non-transparent. The nine LDPC codes defined here are systematic. All are non-transparent[1], and phase ambiguities are resolved using frame markers, which are required for Codeblock synchronization.

Like turbo codes, LDPC codes may be used to obtain greater coding gain than those provided by concatenated coding systems. In contrast to turbo codes, LDPC codes offer the prospect of much higher decoding speeds, via highly parallelized decoder structures. The current recommendation includes turbo codes of rates 1/2 and lower, and LDPC codes of rates 1/2 and higher, so rate 1/2 is the only rate at which either type of recommended code is available.

NOTES

1. LDPC coding, by itself, cannot guarantee sufficient bit transitions to keep receiver symbol synchronizers in lock. Therefore, the Pseudo-Randomizer defined in Section 6 is required unless the system designer verifies that sufficient symbol transition density is assured by other means when the Randomizer is not used.

2. While providing outstanding coding gain, LDPC codes generally may still leave some residual errors in the decoded output. For this reason, when CCSDS Transfer Frames or Virtual Channel Data Units are used, references [1] and [2], respectively, require that a cyclic redundancy check (CRC) be used to validate the frame.

3 Specification

An LDPC code is specified using $v \times w$ parity-check matrix H consisting of v linearly independent rows. A discussion of the corresponding encoder derivation is provided in a later section. A coded sequence of w bits must satisfy all v parity-check equations corresponding to the v rows of H. Parity-check matrices may include additional linearly dependent rows without changing the code. An encoder maps an input frame of $k \leq w - v$ information bits uniquely into a codeblock of $n \leq w$ bits. If $n < w$, the remaining $w - n$ code symbols are punctured and are not transmitted. If $k < w - v$, the remaining dimensions of the code remain unused.

The recommended codeblock lengths n and information block lengths k, and the corresponding rates $r = k/n$, are shown in Table 1 for the suite of recommended LDPC codes. The LDPC code rates r are exactly as indicated in Table 1, unlike the case of turbo codes for which the precise code rates are slightly lower than the corresponding nominal rates due to termination bits.

[1] Differential encoding (i.e., NRZ-M signaling) after the LDPC encoder is not recommended since soft decoding would require the use of differential detection with considerable loss of performance. Differential encoding before the LDPC encoder cannot be used because the LDPC codes recommended in this document are non-transparent. This implies that phase ambiguities have to be detected and resolved by the frame synchronizer.

Information block length k	Code block length n		
	rate 1/2	rate 2/3	rate 4/5
1024	2048	1536	1280
4096	8192	6144	5120
16384	32768	24576	20480

Table 1: Codeblock Lengths for Supported Code Rates (Measured in Bits)

Information block length k	Submatrix size M		
	rate 1/2	rate 2/3	rate 4/5
1024	512	256	128
4096	2048	1024	512
16384	8192	4096	2048

Table 2: Values of submatrix size M for supported codes

For each (n, k) in Table 1, this recommendation specifies the recommended parity-check matrix H.

4 Parity check matrices

The H matrices are constructed from $M \times M$ submatrices, where the submatrix size is listed in Table 2.

4.1 Rate 1/2, 2/3, 3/4, and 4/5 codes

The H matrices for the recommended rate-1/2 codes are specified as follows.

$$H_{1/2} = \begin{bmatrix} 0_M & 0_M & I_M & 0_M & I_M \oplus \Pi_1 \\ I_M & I_M & 0_M & I_M & \Pi_2 \oplus \Pi_3 \oplus \Pi_4 \\ I_M & \Pi_5 \oplus \Pi_6 & 0_M & \Pi_7 \oplus \Pi_8 & I_M \end{bmatrix}$$

where I_M and 0_M are the $M \times M$ identity and zero matrices, respectively, and Π_1 through Π_8 are permutation matrices. The H matrices for the recommended rate-2/3 and rate-4/5 codes are specified with augmented columns and permutation matrices as follows. An H matrix for rate-3/4 is also specified since this rate naturally occurs via the column extension required to achieve rate-4/5.

$$H_{2/3} = \left[\begin{array}{cc|c} 0_M & 0_M & \\ \Pi_9 \oplus \Pi_{10} \oplus \Pi_{11} & I_M & H_{1/2} \\ I_M & \Pi_{12} \oplus \Pi_{13} \oplus \Pi_{14} & \end{array}\right]$$

$$H_{3/4} = \left[\begin{array}{cc|c} 0_M & 0_M & \\ \Pi_{15} \oplus \Pi_{16} \oplus \Pi_{17} & I_M & H_{2/3} \\ I_M & \Pi_{18} \oplus \Pi_{19} \oplus \Pi_{20} & \end{array}\right]$$

$$H_{4/5} = \left[\begin{array}{cc|c} 0_M & 0_M & \\ \Pi_{21} \oplus \Pi_{22} \oplus \Pi_{23} & I_M & H_{3/4} \\ I_M & \Pi_{24} \oplus \Pi_{25} \oplus \Pi_{26} & \end{array}\right]$$

Permutation matrix Π_k has non-zero entries in row i and column $\pi_k(i)$ for $i \in \{0, \ldots, M-1\}$ and

$$\pi_k(i) = \frac{M}{4}((\theta_k + \lfloor 4i/M \rfloor) \bmod 4) + (\phi_k(\lfloor 4i/M \rfloor, M) + i) \bmod \frac{M}{4}$$

where the functions θ_k and $\phi_k(j, M)$ are defined in Tables 3 and 4. Values defined in these tables describe $\phi_k(j, M)$'s using 7-tuples where consecutive positions in the tuple correspond to submatrix sizes from the set M={128 256 512 1024 2048 4196 8192}. The permutation matrix descriptions in conjunction with Tables 3 and 4 describe 28 codes, one for each rate r = {1/2, 2/3, 3/4, 4/5} and M = {128, 256, 512, 1024, 2048, 4096, 8192}. Of these 28 codes, 9 are selected based on criteria provided in section 1 [2]. For any of the H matrices constructed per this description the last M codesymbols are to be punctured (not transmitted).

For example, the parity check matrix for the $(n = 1536, k = 1024)$ code is shown in Figure 1 with dots representing each of the non-zero entries, and its structure is indicated by gridlines (minor gridlines (not shown) spaced at 1/4 the separation of the shown major gridlines also delimit code structure).

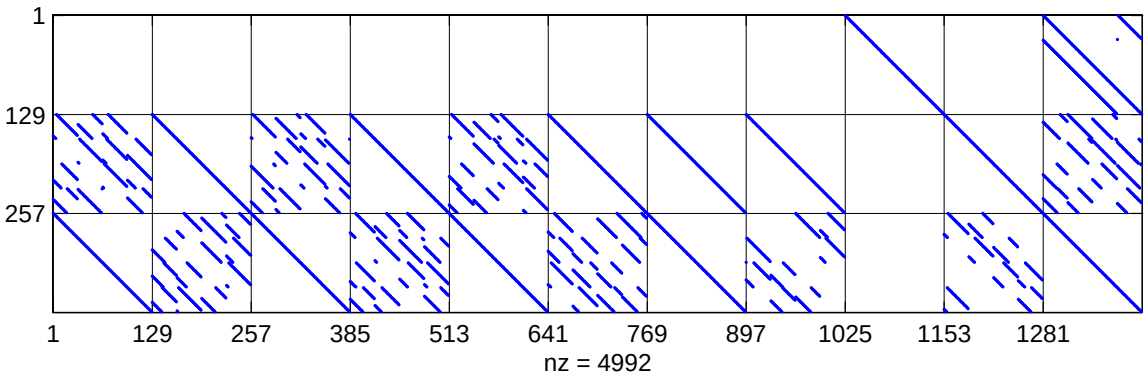


Figure 1: An H Matrix for the $(n = 1280, k = 1024)$ rate-4/5 code. For this code M=128 and columns 1281 through 1408 are punctured.

5 Randomization and Synchronization

LDPC coding, by itself, cannot guarantee sufficient bit transitions to keep receiver symbol synchronizers in lock. Therefore, the Pseudo-Randomizer defined in Section 6 of CCSDS recommendation 101.0-B-6, "Telemetry Channel Coding", is required unless the system designer verifies that sufficient symbol transition density is assured by other means when the Randomizer is not used.

[2] Spacings of $\sim$ 1dB for different rates and 0.6 dB for different lengths

Table 3: Description of $\phi_k(0, M)$ and $\phi_k(1, M)$.

k	θ_k	$\phi_k(0, M)$ $M = 2^7 \dots 2^{13}$	$\phi_k(1, M)$ $M = 2^7 \dots 2^{13}$
1	3	1 59 16 160 108 226 1148	0 0 0 0 0 0 0
2	0	22 18 103 241 126 618 2032	27 32 53 182 375 767 1822
3	1	0 52 105 185 238 404 249	30 21 74 249 436 227 203
4	2	26 23 0 251 481 32 1807	28 36 45 65 350 247 882
5	2	0 11 50 209 96 912 485	7 30 47 70 260 284 1989
6	3	10 7 29 103 28 950 1044	1 29 0 141 84 370 957
7	0	5 22 115 90 59 534 717	8 44 59 237 318 482 1705
8	1	18 25 30 184 225 63 873	20 29 102 77 382 273 1083
9	0	3 27 92 248 323 971 364	26 39 25 55 169 886 1072
10	1	22 30 78 12 28 304 1926	24 14 3 12 213 634 354
11	2	3 43 70 111 386 409 1241	4 22 88 227 67 762 1942
12	0	8 14 66 66 305 708 1769	12 15 65 42 313 184 446
13	2	25 46 39 173 34 719 532	23 48 62 52 242 696 1456
14	3	25 62 84 42 510 176 768	15 55 68 243 188 413 1940
15	0	2 44 79 157 147 743 1138	15 39 91 179 1 854 1660
16	1	27 12 70 174 199 759 965	22 11 70 250 306 544 1661
17	2	7 38 29 104 347 674 141	31 1 115 247 397 864 587
18	0	7 47 32 144 391 958 1527	3 50 31 164 80 82 708
19	1	15 1 45 43 165 984 505	29 40 121 17 33 1009 1466
20	2	10 52 113 181 414 11 1312	21 62 45 31 7 437 433
21	0	4 61 86 250 97 413 1840	2 27 56 149 447 36 1345
22	1	19 10 1 202 158 925 709	5 38 54 105 336 562 867
23	2	7 55 42 68 86 687 1427	11 40 108 183 424 816 1551
24	1	9 7 118 177 168 752 989	26 15 14 153 134 452 2041
25	2	26 12 33 170 506 867 1925	9 11 30 177 152 290 1383
26	3	17 2 126 89 489 323 270	17 18 116 19 492 778 1790

Table 4: Description of $\phi_k(2, M)$ and $\phi_k(3, M)$.

k	θ_k	$\phi_k(2, M)$ $M = 2^7 \dots 2^{13}$	$\phi_k(3, M)$ $M = 2^7 \dots 2^{13}$
1	3	0 0 0 0 0 0 0	0 0 0 0 0 0 0
2	0	12 46 8 35 219 254 318	13 44 35 162 312 285 1189
3	1	30 45 119 167 16 790 494	19 51 97 7 503 554 458
4	2	18 27 89 214 263 642 1467	14 12 112 31 388 809 460
5	2	10 48 31 84 415 248 757	15 15 64 164 48 185 1039
6	3	16 37 122 206 403 899 1085	20 12 93 11 7 49 1000
7	0	13 41 1 122 184 328 1630	17 4 99 237 185 101 1265
8	1	9 13 69 67 279 518 64	4 7 94 125 328 82 1223
9	0	7 9 92 147 198 477 689	4 2 103 133 254 898 874
10	1	15 49 47 54 307 404 1300	11 30 91 99 202 627 1292
11	2	16 36 11 23 432 698 148	17 53 3 105 285 154 1491
12	0	18 10 31 93 240 160 777	20 23 6 17 11 65 631
13	2	4 11 19 20 454 497 1431	8 29 39 97 168 81 464
14	3	23 18 66 197 294 100 659	22 37 113 91 127 823 461
15	0	5 54 49 46 479 518 352	19 42 92 211 8 50 844
16	1	3 40 81 162 289 92 1177	15 48 119 128 437 413 392
17	2	29 27 96 101 373 464 836	5 4 74 82 475 462 922
18	0	11 35 38 76 104 592 1572	21 10 73 115 85 175 256
19	1	4 25 83 78 141 198 348	17 18 116 248 419 715 1986
20	2	8 46 42 253 270 856 1040	9 56 31 62 459 537 19
21	0	2 24 58 124 439 235 779	20 9 127 26 468 722 266
22	1	11 33 24 143 333 134 476	18 11 98 140 209 37 471
23	2	11 18 25 63 399 542 191	31 23 23 121 311 488 1166
24	1	3 37 92 41 14 545 1393	13 8 38 12 211 179 1300
25	2	15 35 38 214 277 777 1752	2 7 18 41 510 430 1033
26	3	13 21 120 70 412 483 1627	18 24 62 249 320 264 1606

Codeblock synchronization is achieved by synchronization of an Attached Sync Marker associated with each LDPC Codeblock. The Attached Sync Marker (ASM) is a bit pattern specified in Section 5 of CCSDS recommendation 101.0-B-6, "Telemetry Channel Coding", as an aid to synchronization, and it precedes the LDPC Codeblock. Frame synchronizers should be set to expect a marker at a recurrence interval equal to the length of the ASM plus that of the LDPC codeblock. All codes in the LDPC family use the 64 bit ASM.

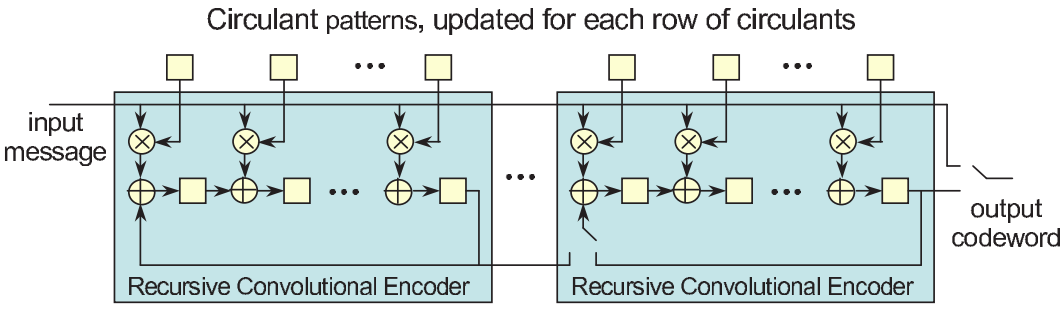


Figure 2: A Quasicyclic Encoder Using Feedback Shift Registers

6 Encoders

The recommended method for producing codeblocks consistent with these H matrices is to perform matrix multiplication by (dense) block-circulant generator matrices G. Note that the family of AR4JA codes supports rates $K/(K+2)$, where $K = 2$ for a rate 1/2 code, $K = 4$ for rate 2/3, and $K = 8$ for rate 4/5. Including circulant dimensions, G will be of size $MK \times M(K+3)$ if punctured columns are included, or $MK \times M(K+2)$ if punctured columns are omitted. Generator matrices G may be constructed as follows.

1. Let P be the $3M \times 3M$ submatrix of H consisting of the last $3M$ columns. Let Q be the $3M \times MK$ submatrix of H consisting of the first MK columns.
2. Compute $W = (P^{-1}Q)^T$, where the arithmetic is performed modulo-2.
3. Construct the matrix $G = \left[\begin{array}{cc} I_{MK} & W \end{array} \right]$, where I_{MK} is the $MK \times MK$ identity matrix and W is a dense matrix of circulants with size $MK \times 2M$.

The matrix G is block-circulant, and is composed of circulants of size $M/4$. Even though simplified by the block-circulant structure [6], computing the matrix inverse in the second step is computationally demanding.

As an example table 5 lists every $M/4$'th row of W (the first row of each set of circulants) in hexadecimal for the generator matrix of the $(n = 1280, k = 1024)$ rate-4/5 code. Note that the last M columns for the punctured symbols are not included.

Encoding of message m requires computing mG. Because G is block-circulant, this can be performed in an efficient bit-serial manner using $4n/M$ linear feedback shift registers, each of length $M/4$, as shown in Figure 2. Initially, the binary pattern from the first row of circulants is placed in the top row of small boxes in the figure. With the switches set as drawn, the k message bits are fed through the encoder one at a time, and the registers are updated and shifted once per bit. After each set of $M/4$ message bits are processed, the circulant patterns are updated for the next row of circulants. Then the switches are changed and the contents of the registers are read out sequentially as the parity portion of the codeword.

6.1 Encoding Example : Rate-1/2 $k = 32$

To clarify the procedure for computing W, we describe a step by step method for finding the generator matrix for a rate 1/2 AR4JA LDPC code with a very short block length (k=32). Consider the protograph of rate 1/2 AR4JA LDPC code as shown in Figure 3.

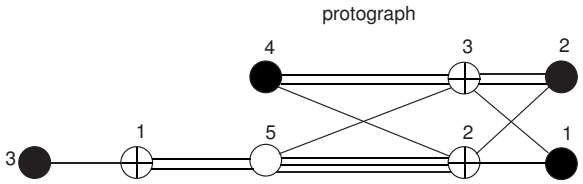


Figure 3: Protograph of rate 1/2 AR4JA LDPC code.

Expand the protograph by factor 4 to remove parallel edges. Assign circulant permutations to edges on the expanded graph. The expanded protograph has the following H matrix. The first 4 rows correspond to check number 1 in Figure 3, the second 4 rows and the last 4 rows correspond to checks 2, and 3 respectively. The first 4 columns correspond to variable node number 1 in Figure 3. The subsequent group of 4 columns correspond to variable node numbers 2, 3, 4 and 5 respectively in Figure 3. For this example $M = 16$ and the $m \times m$ Circulant Permutations have size m=4. Each nonzero entry x^i in the parity check matrix H represents a circulant permutation which is an $m \times m$ identity matrix where each row is circularly shifted to the right by i.

$$H = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & x^1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & x^2 & x^2 & x^2 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & x^0 & x^0 & x^3 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & x^2 & 0 & x^3 & x^1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & x^2 & x^2 & 0 & x^2 \\
1 & 0 & 0 & 0 & 0 & 0 & x^3 & x^2 & 0 & 0 & 0 & 0 & x^3 & x^3 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & x^0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & x^3 & x^3 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & x^3 & x^0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x^3 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & x^0 & x^2 & 0 & 0 & 0 & 0 & 0 & x^1 & 0 & 0 & x^2 & 0 & 0 & 0 & 1
\end{bmatrix} \tag{1}$$

All operations are over the ring of polynomials with coefficients in $GF(2)$ with maximum degree $m-1$ modulo $x^m + 1$.
Step 1. Denote the first 8 columns of H with Q.

$$Q = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & x^3 & x^2 \\
0 & 1 & 0 & 0 & x^0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & x^3 & x^0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & x^0 & x^2 & 0
\end{bmatrix} \tag{2}$$

Denote the last 12 columns of H with P.

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & x^1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & x^2 & x^2 & x^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & x^0 & x^0 & x^3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & x^2 & 0 & x^3 & x^1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & x^2 & x^2 & 0 & x^2 \\ 0 & 0 & 0 & 0 & x^3 & x^3 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & x^3 & x^3 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x^3 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & x^1 & 0 & 0 & x^2 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Step 2. Find Cofactor Matrix of P , denote it by $P_{2,c}$.
Step 3. Find determinant of P_2 , denote it by polynomial $d(x)$. In this example $d(x) = x + x^2 + x^3$.
Step 4. Find the inverse of polynomial $d(x)$ using Euclid's algorithm. In this example the inverse is $d^{-1}(x) = x + x^2 + x^3$.
Step 5. Multiply the inverse polynomial $d^{-1}(x)$ by Cofactor matrix $P_{2,c}$, denote the result by $P_{2,i}$.
Step 6. Multiply the $P_{2,i}$ by Q , denote the result by W_2.
Step 7. Find the reciprocal polynomial of each entry of W_2 and denote the result by W.
Step 8. The generator matrix G is obtained as $G = [IW]$, where I here is an 8×8 identity matrix.
Step 9. The last 4 columns of W corresponds to punctured nodes i.e. not transmitted nodes. Thus for encoding we need only the first 8 columns of W which will be denoted by W_P. The martix W_P for our simple example is given as:

$$W_P = \begin{bmatrix} 1+x^2 & x^2+x^3 & 1 & x & 1+x^3 & x^3 & 1+x+x^2 & 0 \\ 1+x^2+x^3 & 1+x & 1+x^2 & x^3 & x+x^2 & 0 & 1+x+x^2 & x^2 \\ 1+x^2+x^3 & x+x^2+x^3 & x+x^2 & 0 & 1+x^2+x^3 & 1+x^2 & x+x^3 & 1+x^2+x^3 \\ 1+x & x & 1+x+x^2 & x+x^2 & x+x^2+x^3 & x^3 & 1+x+x^2+x^3 & x+x^2 \\ 0 & x^2+x^3 & 0 & x+x^3 & 1+x^3 & x^3 & 1+x+x^2 & x \\ x^2+x^3 & 1+x & 1+x^2 & 1+x & x^2 & x+x^2 & 1+x^2+x^3 & x^3 \\ x+x^2 & x+x^3 & 1+x^3 & x^2+x^3 & 1+x^2+x^3 & 1+x+x^3 & 1+x^3 & x+x^2+x^3 \\ x^2+x^3 & 0 & 0 & 1+x+x^2+x^3 & x^3 & 1+x+x^3 & x & x+x^2 \end{bmatrix} \tag{4}$$

The parity generation polynomials in W_P also can be expressed as hexadecimal digits as shown below.

$$W_P = \begin{bmatrix} 5 & C & 1 & 2 & 9 & 8 & 7 & 0 \\ D & 3 & 5 & 8 & 6 & 0 & 7 & 4 \\ D & E & 6 & 0 & D & 5 & A & D \\ 3 & 2 & 7 & 6 & E & 8 & F & 6 \\ 0 & C & 0 & A & 9 & 8 & 7 & 2 \\ C & 3 & 5 & 3 & 4 & 6 & D & 8 \\ 6 & A & 9 & C & D & B & 9 & E \\ C & 0 & 0 & F & 8 & B & 2 & 6 \end{bmatrix} \tag{5}$$

6.2 Encoding Example : Rate-4/5 $k = 1024$

The method described in previous section is used to obtain the generator matrix for one of the proposed codes. Here we show the derived W_P matrix in table 5 for Rate-4/5 k=1024,

Table 5: Description of Rate-4/5 k=1024 Quasi-Cyclic Encoding. Degree-32 parity generation polynomials expressed as hexadigit characters.

	Col Ind							
Row Ind	0,31	32,63	64,95	96,127	128,159	160,191	192,223	224,255
0,31	8AD371E6	3AB8417F	D242FA5F	55E49AAF	C896417C	30D2074C	D46111F2	F74C2C01
32,63	A46E4D42	8785B1F9	D8B4E21F	29CBC29B	AEA86494	1632C25D	592CF718	233853E4
64,95	E9C00937	C6052526	652373D9	F0E5DA0B	87DAA8E7	83390FFE	9A047476	A7E6A8FE
96,127	3808686A	D4706057	D160CE6D	D1FBDC49	BC2C9D9D	15C63920	7F397CCC	B46CD901
128,159	32C682B9	5BE87202	4F8682DC	7499AC4A	D1D2D257	873D0962	856C5B9F	8DD9C268
160,191	AD2A4EBA	D1E92DA8	244A0B00	209BDE08	1A7EBA13	A58F8A20	E918B977	9F65BC89
192,223	86F98142	53CEEC2C	54926588	FC2F4C12	B7883921	F2617F0F	0020E433	766B64E9
224,255	22547791	E6F83A16	68276FB8	F74FDE0A	DF439AE1	FD54684A	DFCB86D4	2310E119
256,287	9057F152	2AC4A2CA	CE747CC5	884178E4	A746FB68	2F81FBC0	F0BD1211	EACFBA9F
288,319	CDE507D9	D76A4E86	2DD0B259	985C5C7F	79BC655F	18914CA6	AC5D996B	07F67B32
320,351	A7F8D121	F651AD50	8FE0E82C	D2E26CF7	A5E5C46F	5D937AC5	76D90924	1915E526
352,383	75DCFBBD	645F404E	EA309F61	04F99C05	8C59D4E9	75A24DE1	1CC5A307	9B559A92
384,415	E4364EF3	59421CDE	2243EC5D	DBE8EE77	53993A27	DC807C7E	253001DE	E74FE4A6
416,447	41EC63AE	EC802338	68F9678A	FAA8D282	7B2698FB	EF8E3C8F	22017976	61373A36
448,479	1FB654F6	41FD4FF9	AACA46CB	81966224	83528FAB	57524387	C1B0A115	048A3C5E
480,511	C3B5D55D	B660F378	12C1CA6E	2AB2BA3F	F362B5C2	348042CC	3B47D7C6	71F74C1B
512,543	8F413767	41E7F552	1C791B28	402C13C4	FD6B12D1	591DC413	646AC516	8487F917
544,575	8CC55DD2	93683704	607D5B56	B65BC01B	82B9133F	1708DEA7	280FFC33	6042EDB2
576,607	61ABFC0A	89D16F4A	CFB96BE9	726E3948	EFC284D1	5DCF76C5	3691E43E	ADE6F06A
608,639	6709C0EB	57ECCD19	C6C16A91	FB816854	314972D2	39BC3782	4D749BFB	3A13ABA5
640,671	BD110A54	0816D36E	84BC96D3	ABF79703	760A499F	41073994	65AA8809	6855C9D0
672,703	737F7F18	6CAC600D	910E12A9	9DD8CDD2	E9BFCF66	051AD4C7	5E429C54	F97B1B3A
704,735	6861F158	4D4BEFA3	723D4B5A	DB6178FD	5FFDEF35	E9A91ACF	32A6EEE5	26C8BB48
736,767	9A5C03FF	E4A0A23B	8E132DCC	CC57455D	658AE5C9	B274EB9F	FE30AD96	9F8A82EF
768,799	B98A17DE	5F5FF6F5	CE5DB164	31486AC5	347D1820	5A62C258	A6FB6306	051C2470
800,831	A03BA437	15277566	50C054E5	086DF88D	EF5B2EBB	A6AB6F46	ED7572AA	3675EFA8
832,863	509FE28A	803770FF	36699548	8DA0E8E1	0CFFAC97	EA94C762	3F96B62A	60BD851D
864,895	F6570156	60A9458E	F3551EF7	B7AD4AB1	669250F9	716DCD86	69F5E8D2	743414DA
896,927	4E441AAE	9942A768	39D0E0BE	EA80C8D5	4B98636E	AB4700AA	9A2019F8	C1FE39BA
928,959	B85BB056	086D176F	85070BA4	0792E424	60525B7F	6B96B4E0	9E5BE3C0	AACC558A
960,991	0FF7F976	8FD3E9E0	24136C97	AF578393	627E062B	70DEB711	B5068749	B50288CE
992,1023	F53F7F4C	AEDC610E	088F621C	FFBC4723	CAEFCB9E	5F126260	AF4A20C1	A221B79D

where $M = 128$ and 32×32 circulant permutations with $m = 32$ were used. Note that the elements describing the parity portion of the generator are degree-32 polynomials. In table 5 these polynomials are represented with 8 hexadecimal digits. For instance, the $(1, 1)$ element (`0x8AD371E6`) denotes polynomial $x + x^2 + x^5 + x^6 + x^7 + x^8 + x^{12} + x^{13} + x^{14} + x^{16} + x^{17} + x^{20} + x^{22} + x^{23} + x^{25} + x^{27} + x^{31}$.

A Performance

Figure 4 shows the frame error rates (dashed) and symbol error rates (solid) for the k=16384, 4096, 1024 block lengths at rates 1/2,2/3, and 4/5. Within the group of curves for any given color (red, green, or blue) performance monotonically improves with longer block-length. Performance curves for the codes with $k = 1024$ and $k = 4096$ were determined by hardware simulation on a Xilinx Virtex-II FPGA [8]; performance for the $k = 16384$ codes are from software simulations. In each case, a large maximum number of iterations was allowed, and a stopping rule was used so the average number of iterations required remained small.

The performance of these codes is as shown in Figure 4. Data points at the lowest frame-error rates are constructed from at least 2 block decoding errors (given present simulation power, it should be possible to gather at least 10). The majority of points at higher frame-error rates are constructed from 50 block errors.

References

[1] *Telemetry Channel Coding*. Recommendation for Space Data System Standards, CCSDS 101.0-B-6. Blue Book. Issue 6. Washington, D.C.: CCSDS, October 2002.

[2] D. Divsalar, S. Dolinar, J. Thorpe, and C. Jones, "Constructing LDPC codes from simple loop-free encoding modules," *IEEE International Communications Conference*, (Seoul, Korea), May 2005.

[3] D. Divsalar, S. Dolinar, and C. Jones, "Low-rate LDPC codes with simple protograph structure," *IEEE International Symposium on Information Theory*, (Adelaide, Australia), September 2005.

[4] D. Divsalar, C. Jones, S. Dolinar, and J. Thorpe, "Protograph based LDPC codes with minimum distance linearly growing with block size," *IEEE Global Telecommunications Conference*, (Saint Louis, USA), December 2005.

[5] K. Andrews, S. Dolinar, D. Divsalar, and J. Thorpe, "Design of Low-Density Parity-Check (LDPC) Codes for Deep Space Applications," *IPN Progress Report 42-159*, JPL, November 2004.

[6] K. Andrews, S. Dolinar, and J. Thorpe, "Encoders for Block-Circulant LDPC Codes," *IEEE International Symposium on Information Theory*, (Adelaide, Australia), September 2005.

[7] S. Dolinar, D. Divsalar, and F. Pollara, "Code Performance as a Function of Block Size," *IPN Progress Report 42-133*, JPL, May 1998.

[8] C. Jones, E. Valles, M. Smith, and J. Villasenor, "Approximate-Min* Constraint Node Updating for LDPC Code Decoding," *MilCom 2003*, (Boston, MA), October 2003.

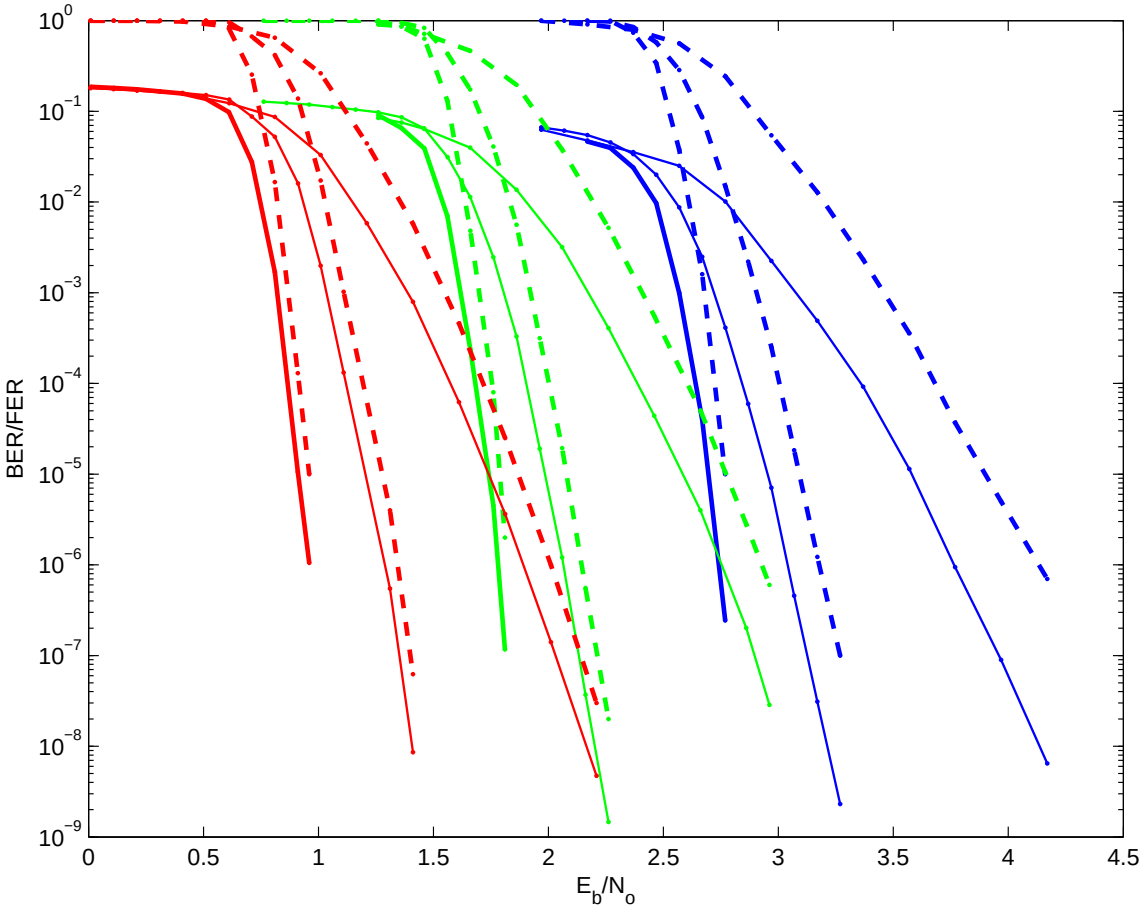


Figure 4: Performance of LDPC Codes: red=rate-1/2, green=rate-2/3, blue=rate-4/5, for $k = 16384, 4096, 1024$.

[9] J. Lee and J. Thorpe, "Memory-Efficient Decoding of LDPC Codes," *IEEE International Symposium on Information Theory*, (Adelaide, Australia), September 2005.